Real-time denoising of ToF measurements by spatio-temporal non-local mean filtering

Citation

Year
2013

Version
Peer reviewed version (post-print)

Link to publication
TUTCRIS Portal (http://www.tut.fi/tutcris)

Published in
2013 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2013, San Jose, CA, USA, 15.-19.7.2013

DOI
10.1109/ICMEW.2013.6618384

Copyright
© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Take down policy
If you believe that this document breaches copyright, please contact tutcris@tut.fi, and we will remove access to the work immediately and investigate your claim.
REAL-TIME DENOISING OF TOF MEASUREMENTS BY SPATIO-TEMPORAL NON-LOCAL MEAN FILTERING

Mihail Georgiev, Atanas Gotchev, Miska Hannuksela
Tampere University of Technology, Tampere, Finland; Nokia Research Center, Tampere, Finland

ABSTRACT
This work addresses the problem of denoising of range data obtained by ToF continuous signal modulation camera working in low-power mode. The proposed approach is based on non-local mean filtering applied over extensive spatio-temporal block search in complex-valued signal domain. The extensive search allows for using shorter integration times of the range sensors and leads to an effective overcomplete structure suitable for denoising. The filter structure is optimized for real-time operation and achieves \( O(1) \) performance for arbitrary patch size by utilizing summed area tables and look-up table data fetching. The experimental results show practically the same performance while compared with state-of-the-art approaches, for greatly improved speed.

Index Terms— Time-of-Flight, ToF, Photonic Mixer Device, PMD, range sensor, denoising, integral summed tables, non-local means

1. INTRODUCTION
A number of imaging applications, such as 3D scene capture and reconstruction, and virtual view synthesis require precise knowledge about the scene depth. Therefore, sensors measuring distances, so-called range sensors as well as the corresponding sensing techniques have emerged recently. A class of range sensing devices uses the so-called Time-of-Flight (ToF) principle, where the distance is measured by computing elapsed time between emitted and reflected signals [1].

A typical ToF device consists of a beamer, an electronic light modulator and a sensor chip. The beamer is made of an array of light-emitting diodes (LED) operating in near-infrared wavelengths (e.g. 850 nm). It radiates a point-source light of a continuously-modulated harmonic signal which illuminates the scene. The reflected light from object surfaces is sensed back by pixels of the sensor chip, which collects pixel charges for some period denoted as integration time \( I_L \). For each pixel, the range data is estimated in relation to phase-delay between sensed signal and the one of the light modulator [1-4]. The phase-delay estimation is performed as a discrete cross-correlation process of several successively captured samples taken between equal intervals during same modulation periods of fixed frequency. Denote the sample data as \( R_n \) (\( n=1,2,...,N \), \( N\geq4 \)). The mixed signal components (amplitude and phase) are estimated from the sampled data as follows:

\[
A = \frac{2}{N} \sum_{n=0}^{N} R_n e^{-j2\pi \frac{n}{N}}, \quad \phi = \arg \left( \sum_{n=0}^{N} R_n e^{-j2\pi \frac{n}{N}} \right),
\]

where \( A \) is the modulation amplitude, and \( j \) is the imaginary unit. The sensed distance - \( D \) is proportional to the phase, while the error in distance measurements is proportional to the square inverse of the amplitude

\[
D \propto \frac{\phi}{4\pi f c_L}, \quad \sigma_D^2 \propto \left( \frac{1}{A^2} \right),
\]

where \( f \) is the frequency of the emitted signal and \( c_L \) is speed of light through dry air (~298.10\(^9\) km/h). The exact value of \( D \) is calculated after precise calibration of the sensor.

While being quite precise in measuring distances in indoor conditions, ToF devices exhibit generic drawbacks related with their principles of operation. Some specific properties of the sensed scene, ambient light conditions or technological limitation of the used device can lead to erroneous range measurements, modeled by a measurement error (noise) \( E_D \) with variance \( \sigma_D^2 \). A specific case of interest is the so-called low-sensing mode, it which the sensor is more restricted, by e.g. requirements for miniaturization leading to limited beamer size, decreased number of LED elements; embedding into portable low-
power devices; requirements for low power consumption, etc. For such a mode the noise presence becomes a dominant problem, which should be addressed by dedicated denoising methods. The case is illustrated in Fig. 1. In the figure, the distance measurement error is plotted against the measured amplitude and curves of mean($E_n$) and var($E_n$) are fitted to illustrate the dependence in Eq. (2). An amplitude threshold can delineate the operating range into low-sensing and normal modes. Below the threshold, the distance measurement error goes above the specifications given in ToF operating manuals [3, 4], where the amplitude can still serve as an estimate of the reliability of range measurements [5, 6]. Examples of depth maps sensed in low-amplitude conditions are given in Fig. 2 where integration time is varied as of $I_f$ 400 µs, 200 µs, 100 µs, and 50 µs when compared to normal operating mode of 2000 µs.

A remedy for such cases is to introduce a denoising procedure applied to the computed distance map in a post-measurement stage. While the influence of noise in distance maps is very high and resembles a moving grainy fog, the distance data is of low-texture content and is usually considered a piece-wise smooth function. This makes the denoising step well tractable in the light of modern non-local denoising approaches.

2. PROPOSED DENOISING ALGORITHM

2.1. Relation to prior work

The core of our approach is the non-local means (NLM) denoising paradigm. The general idea is to find and stack similar blocks (patches) of pixel data together and utilize their similarity measures as weights in a filtering procedure based on averaging. We based our approach on the original NLM [7], however other non-local transform-domain filtering schemes are also possible [8]. The general NLM for a pixel with coordinate $x$ is defined as follows [7]:

$$NLM[x] = \frac{1}{C_N(x)} \int_{\Omega} \exp \left( - \frac{G \times [U(x + .) - f]^2}{h^2} \right), \quad (3)$$

where $C_N$ is a normalization factor, $G$ is a Gaussian kernel, $\Omega$ is the search range for similar patches, $U$ is a pixel map (patch), $h$ is a filter parameter tuned in relation with noise variance, $\int_{\Omega}$ denotes “centered convolution operator”, $(+\cdot)$ denotes the pixel indices of the spatial neighborhood, i.e. the patch with size $B_x$.

We have modified this approach to work for complex-valued patches. More precisely, the patches are formed by the pre-filtered maps of $(A, \phi)$ as $- (A_U, \phi_U)$ given in Eq.(1) and combined into complex numbers $Z$ pixel-wise:

$$\phi_U, A_U \rightarrow Z = A_U \big| e^{i\phi_U} \big|, \quad Z \rightarrow A_U = |Z|, \quad Z \rightarrow \phi_U = \arg(Z). \quad (4)$$

Thus, the modified NLM filter, denoted as $NLM_{CLX}$ is given by:

$$NLM_{CLX}[x] = \frac{1}{C_N(x)} \int_{\Omega} \exp \left( - \frac{G \times |Z(x + .)|^2}{h^2} \right). \quad (5)$$

Fig. 2. Examples of ToF depth capture in low-sensing environment (top-to-bottom, left-to-right): depth ($D$) map in default sensor integration mode ($I_f = 2000\,\mu s$); amplitude ($A$) map in default mode; histogram of $A$; depth maps sensed in low-sensing modes, i.e. for $I_f$ 400 µs, 200 µs, 100 µs, and 50 µs; corresponding amplitude histograms with percentage of erroneous pixels; and corresponding amplitude images.

Fig. 3. Denoising results by $NLM_{CLX}$ for input images with three different integration times (left-to-right) 50 µs, 100 µs, and 200 µs. Top row, input images with PSNR=18.10, 22.4, 26.58 dB; Bottom row denoised images, PSNR=27.94, 31.62, 35.16 dB.

A preliminary filtering technique for the individual components of $A_U$ and $\phi_U$ is performed by some mild smoothing filtering (e.g. Gaussian) before their coupling into a complex number. The reason for this preliminary filtering is to tackle possible structural artifacts. The effect of improvement is achieved when the source exhibits areas of multi reflectivity, low reflectivity and small angle of incidence. For such cases, the preliminary smoothing improves search confidence of block boundaries, the phase wraps-over and provide measures of extreme errors among
closely situated pixels which in fact are sensing similar distances [6]. Some denoising results utilizing complex domain signal filtering by applying NLm_{CLx} are presented in Fig. 3. As seen in the figure, the denoising procedure gives better results compared to longer integration times. This fact points out possibilities for further improvement based on over-complete data representation in spatio-temporal domain.

2.2. Spatio-temporal NLM filtering in complex domain

In this section we propose a modification of our approach [6], targeting real-time implementation with low-complexity fixed-point arithmetic. Nicely enough, the modification brings also further improvements in the performance. The approach is based on similarity search and filtering in spatio-temporal complex-valued signal domain. It makes use of highly overcomplete data structure utilizing subsequent time frames where measurements are taken with shorter integration time. It is less expensive in terms of power consumption to get higher number of noisy frames and to denoise them properly instead of making less-noisy measurements with longer integration time. The modified approach optimizes the similarity search, allowing for example to impose smaller spatial neighborhood range \( \Omega_{nxn} \) with e.g. \( n<7 \).

The non-local approaches require collecting reasonable amount of patches for their joint filtering. The speed of the solutions proposed in [6, 7] depends on the search parameters and content data, and the calculation of the exponent functions in Eq. 3 and Eq. 5. Therefore, our modification specifically addresses the way how spatio-temporal patches are collected in an efficient manner. In the spatio-temporal domain, the search range \( \Omega(x,t) \) becomes a parallelogram scanned in a spiral mode as illustrated in Figure 6. The algorithm is formalized as follows:

1. While Capturing ToF frames
2. Store data maps
   2.1. Store frames in temporal stream buffer
   2.2. Choose a “Reference” frame
3. For each pixel of Reference map DO
   3.1. Define a centered patch
   3.2. Define patch search neighborhood – \( \Omega \)
   3.3. For frames in stream buffer DO
       3.3.1. For each frame DO
           3.3.1.1. Select patches for \( \Omega \)
           3.3.1.2. Estimate filter similarity weight
           3.3.1.3. Accumulate similarity weights
           3.3.1.4. Accumulate filter normalizer
       3.3.2. Average and normalize similarity weights
   3.4. Save denoised result in Denoised Map
4. Store resulted output
5. Update stream buffer
6. Select next frame as Reference
7. Repeat Steps 1 – 5

The block scheme of the application flow is given in Fig. 5.

2.3. Simplified spatio-temporal denoising approach

A direct implementation of Eqs. (2) and (3) requires a convolution step for each block similarity cost. We propose a simplified approach specific for ToF data, based on the assumption that PMD range data forms a piecewise-smooth function and does not contain high-textured content as in color images. Both the temporal-domain neighborhood \( \Omega(t) \) and the block size \( B_S \) can be made smaller taking advantage of the higher number of accumulated blocks. The small size of the PMD sensor (e.g. 204x204) implies more available memory when compared to 2D HD memory format (1920x1080).

In our approach, we take the Gaussian convolution out of the weighted averaging operation. An additional data map denoted as “similarity map” is created by prefiltering of
original complex (noisy) data $Z$ with a Gaussian kernel. The prefiltered map is used for finding the similarities and forming the weights for the noisy data.

$$NLM_{CLX}[x,t] = \frac{1}{C_N(x)} \left( \int_{\Omega} \exp \left( -\frac{|Z_p(x) + G \cdot t|}{\exp(\frac{h}{2})} \right) \right)$$

2.4. Summed area tables

A Summed Area Table (SAT) (a.k.a. Integral Image Table) is a data structure, which allows for quick generation of local sums for arbitrary sized rectangular data blocks [9, 10]. SATs can be used for fast calculation of cumulative linear functions [11] block-wise for the expense of extensive memory use. As the name suggests, the pixel value is a data structure, which allows for quick generation of local sums for arbitrary sized rectangular data blocks [9, 10].

A Summed Area Table (SAT) (a.k.a. Integral Image Table) is a data structure, which allows for quick generation of local sums for arbitrary sized rectangular data blocks [9, 10]. SATs can be used for fast calculation of cumulative linear functions [11] block-wise for the expense of extensive memory use. As the name suggests, the pixel value is a data structure, which allows for quick generation of local sums for arbitrary sized rectangular data blocks [9, 10].

For a rectangular patch with size $B$, with vertices defined by the positions of four pixels – $p_1, p_2, p_3, p_4$ (c.f. Fig 7), the sum of pixels within $B$ is calculated by the difference of vertex pixels: $p_3 + p_1 - p_4 - p_2$, which computationally can be realized through a look-up table.

We implement the similarity search through SATs. More specifically, for all shifts determined by the search range structural element $O$, we calculate a similarity cost function (CF) in the form of pixel-wise squared difference. Each CF image is turned to an SAT and the local summation determined by the position of the reference block is calculated over the corresponding SATs. The use of SATs allows computing CF for $O(1)$ operations.

The procedure is depicted in Fig. 7 and implemented as follows:

1. Select map $R$ as reference
2. For each frame $- T$ DO
   2.1. For all pixel shifts of $T$ in neighborhood $- \Omega$
       DO
       2.1.1. Shift $T$, Store shifted map $T_\delta$
       2.1.2. Apply CF per-pixel between $R$ and $T_\delta$
       2.1.3. Store map of differences $M_\delta$
       2.1.4. Create and store SAT of $M_\delta$
   2.2. END
3. END

3. EXPERIMENTS AND RESULTS

3.1. Test setup

Our experimental setup includes a captured scene where the relative object position to camera is kept unchanged, but the sensing conditions are varied by changing sensor operational parameters. The testing scene is designed from planar objects facing frontally the camera, but placed in arbitrary positions. The objects are made from materials of different reflection or painted in different color textures. A PMD Vision CamCube 2.0 device has been used in the experiments [3]. The default camera capturing settings and integration time of 2000 $\mu s$ have been set to model normal sensing conditions (i.e. amplitude confidence ensuring measuring error smaller than specified for the device). We have prepared Ground Truth (GT) images by averaging 200 consecutively captured frames and manually fine tuning of planar depth values. We model the varying sensing environment by decreasing the integration times for capturing raw samples within the rage 50÷400 $\mu$s (cf. Fig. 2). For each experiment integration time, Table 1 lists the percentage of pixels which amplitudes $A$ are below the low-sensing threshold for this ToF device. The following methods have been implemented and tested: classical Non-local Means [7] - $NLM$ (Eq.5), our proposed modification of $NLM$ for complex data [6] - $NLM_{CLX}$ (Eq.7), an $NLM$

\begin{align*}
NLM_{CLX}[x,t] &= \frac{1}{C_N(x)} \left\{ \int_{\Omega} \exp \left( -\frac{|Z_p(x) + G \cdot t|}{\exp(\frac{h}{2})} \right) \right\}
\end{align*}

Fig. 8. Denoising of single frame for different noisy conditions ($I_\delta$) from left to right: $\Omega_{[5x5]}$, $B_{[7x7]}$, $\Omega_{[5x5]}$, $B_{[7x7]}$.

Fig. 9. Visualization of denoised depth map (single frame) for $I_\delta=50$ $\mu$s, $\Omega_{[5x5]}$, $B_{[7x7]}$. From top to bottom and from left to right: GT data; noisy input, visual scene; $NLM$; $NLM_{CLX}$; $NLM_{ADA}$; $NLM_{INT}$; $NLM_{CLX,INT}$; $NLM_{ADA,INT}$.

version - $NLM_{ADA}$ where the filter parameter $h$ is adapted to $E_0$:

\begin{align*}
NLM_{ADA}[x] &= \frac{1}{C_N(x)} \left\{ \int_{\Omega} \exp \left( -\frac{G \cdot |U(x)|}{\exp(\frac{h}{2})} \right) \right\}
\end{align*}
3.2. Experiments

We have tested the performance and speed of the methods versus sizes of search range and block. The comparisons are made with respect to PSNR calculated as follows:

\[ MSE = \frac{1}{N_p} \sum_{x=0}^{x=N_p} (D_C(x) - D_G(x))^2 \times PSNR[dB] \]  

(10)

3.3. Single-frame denoising

The selected methods were compared for varying noise influence modeled by varying integration time \( I_T \in [50 \mu s \div 400 \mu s] \). Two combinations of search range \( \Omega \) and block size \( B_S \) are employed as illustrated in Fig. 8. The performance of SAT-optimized version is comparable with the original methods. Visual results given in Fig. 9 for an extreme noise presence \( (I_T = 50 \mu s) \) show non-discriminable differences, except for a slight increase in sensed artifact areas (near object edges)\[6\]. Visually, the best performance is demonstrated by NLM\_CLX, where suppression of such artifacts is visible for both classical and speed-optimized versions. The second test, illustrated by Fig. 10 shows the denoising performance, when filter parameters - \( h \) (or \( h_i \)) are tuned for optimal performance for different \( \Omega_{\text{MAX}} \), \( B_{\text{MAX}} \). The results suggest that NLM\_CLX and its optimized version - NLM\_CLX\_INT perform always better for any patch size selection parameters. The plots instruct that those sizes can be kept relatively small (e.g. \(<[5x5]\)) for optimal performance.

3.4. Multiple-frame denoising

The denoising methods NLM, NLM\_ADA, and NLM\_CLX were implemented with patch selection based on spatio-temporal search, i.e. having a three-dimensional search region \( \Omega \). The data for spatio-temporal patch selection is taken from sequences (video) of 40 frames for different sensor integration times – \( I_T \). The experiment was aimed at testing the performance of speed-optimized versions of denoising solutions only. Table 1 summarizes the results for varying integration times and different sizes of \( \Omega \) and \( B_S \).

### Table 1. Denoising performance of SAT-optimized methods.

<table>
<thead>
<tr>
<th>Size parameters ([n \times n])</th>
<th>Optimized approaches</th>
<th>( B_S )</th>
<th>( \Omega )</th>
<th>( B_S )</th>
<th>( \Omega )</th>
<th>( B_S )</th>
<th>( \Omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Omega )</td>
<td>NLM_CLX</td>
<td>NLM_CLX_INT</td>
<td>NLM_ADA</td>
<td>NLM_INT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Noise ( I_T \in [50 \mu s \div 400 \mu s] )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>38.09</td>
<td>38.38</td>
<td>37.75</td>
<td>38.09</td>
<td>38.52</td>
<td>38.53</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38.14</td>
<td>38.18</td>
<td>37.66</td>
<td>38.94</td>
<td>39.07</td>
<td>38.63</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38.06</td>
<td>38.18</td>
<td>37.95</td>
<td>38.92</td>
<td>38.64</td>
<td>38.67</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>38.80</td>
<td>39.15</td>
<td>38.67</td>
<td>38.80</td>
<td>39.15</td>
<td>38.67</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38.80</td>
<td>39.15</td>
<td>38.67</td>
<td>38.80</td>
<td>39.15</td>
<td>38.67</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38.80</td>
<td>39.15</td>
<td>38.67</td>
<td>38.80</td>
<td>39.15</td>
<td>38.67</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38.80</td>
<td>39.15</td>
<td>38.67</td>
<td>38.80</td>
<td>39.15</td>
<td>38.67</td>
</tr>
</tbody>
</table>

The performance is tracked over the number of frames as plotted in Fig. 11. Fig. 12 shows visual results for the case of strong noise influence \( (I_T = 50 \mu s) \). The plot of Fig. 11. represents an empirical evidence that best obtained results starts to converge for relatively small number of frames (e.g. 8 frames) which is instructive for optimal memory size of data stream buffers and decrease of frame-processing time. While processing frames in spatio-temporal domain, the result is always better compared to single-frame denoising for the same \( I_T \). In the figure, one can see that the performance for 4 and 8 processed frames when \( I_T = 50 \mu s \) is higher compared to processing of single frame captured with...
I$_T$=100 μs and I$_T$=200 μs (c.f. Fig. 11.). The results in Table 1 quantify the superior performance of NLM$^{CLX(INT)}$.

Visual depictions in Fig. 12 show also nice reconstruction of depth discontinuities. One can notice good edge preservation and artifact suppression and a very close resemblance to GT data.

### 3.5. Constant $O(1)$ complexity and computational costs

The similarity CF in the denoising approaches is calculated by look-up table (Eq. 7). Therefore, it is expected that SAT-optimized versions will perform in constant speed - $O(1)$. This is demonstrated in Fig. 13 for different block sizes of $B_S$. The performance is indeed constant with a slight delay in speed, when the $B_S$ size increases. It is explained by the fact that for bigger sizes, the data maps are increased to meet boundary conditions and to avoid non-block covered areas.

The speed performance achieved in our code realization (Intel Core 2 Duo T6800, single core used) on MS Visual Studio 2010 C++ compiler in default settings is $\sim$1260 Hz (patches per second) of arbitrary block size $B_S$. It is well above the required real-time speed.

The most computationally intensive calculations are left for calculating the exponent function in Eq. 6. The input raw data can be quantized, without loss of precision (e.g. $>$0.5 cm [3, 4]), as 16-bit numbers. The exponent function can be pre-computed by a look-up table and further quantized. Such optimization boosts the calculation speed for this particular step by about 5÷10 percent when floating-point processing unit is used and about 200÷400 percent for fixed-point processing unit (e.g. Texas Instruments - OMAP4470 ARM-Cortex-A9v7 SIMD NEON).

Utilized memory of SAT for $\Omega_{\{5x5\}}$ pixel region requires 204x204x5x5x2 bytes i.e. $\sim$2Mb which is smaller than an RGB color image of HD size format (1920x1080x3, $\sim$6mb).

### 3.7. Conclusions

We have presented a ToF data denoising approach optimized for real-time operation. The approach is based entirely on software post-processing and does not require hardware changes of current ToF technology. The solution is well scalable for different memory and data processing demands and achieves very good results for very extreme cases of noise presence even when only a few consecutive frames are used. Our prototype application shows very good results (cf. Table 1) and code structure has easy portability up to very low-complexity computation systems, since it does not require floating-point processing units, because it is almost entirely based on pre-computed look-up tables and SAT-based block search.

### 5. REFERENCES

(1st row): GT data; noise input; averaged result, (top-to-bottom): Ω[3x3], B[3x3], Ω[3x3], B[7x7], Ω[7x7], B[5x5]; left to right: NLM; NLMCLX; NLMADA.

Fig. 13. Speed comparison plots for $O(1)$ performance demonstration for different $B_s$ (left-to-right): a) non-log scale plot, b) log scale plot.


